Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institut für
Technische Informatik und
Kommunikationsnetze

Dynamic Visualization of Geophysical Data

Semester Thesis

Feiyu Jia

jiaf@student.ethz.ch

Computer Engineering and Networks Laboratory
Department of Information Technology and Electrical Engineering
ETH Zürich

Supervisors:
Matthias Meyer
Dr. Samuel Weber
Prof. Dr. Lothar Thiele

January 29, 2019

Acknowledgements

I would like to thank my supervisors Matthias Meyer and Samuel Weber for their amazing ideas and technical guidance during the period of this project. I also want to thank my family and friends for their support whenever I needed it.

Abstract

In order to help get a better understanding of the geophysical data collected through hundreds of sensors deployed in Swiss Alps, a multi-scale, multi-faceted visualization tool is created for PermaSense. In this work, long-term geophysical data is able to be plotted with the help of suitable downsample methods. Besides, tools to visualize Fast Fourier Transform and Spectrogram are added in this interface to help analyze the seismic stream.

Contents

Chapter 1

Introduction

It is known that rock falls is one of destructive natural hazards [1] and it would be helpful if the potential for rock falls can be assessed. It turns out that most rock falls are associated with some triggering events, such as earthquakes, rainstorms, or periods of warming producing a rapid melting of snow [2, 3]. Human beings will be in a position of advantage if the probability of these triggering events can be assessed from different types of geophysical data.

Wireless sensor networks (WSN) with a range of different sensors can be used to analyze geophysical processes. Currently, a system called *PermaSense* [4] is deployed in the Swiss Alps, which has several field sites with multi-year time series. Besides the long time frame of data, there are many different sensors collecting different types of data. In order to help understand geophysical dataset more efficiently, a multi-scale and multi-faceted visualization tool is required to show these data dynamically and help observe its intricate property.

The current interface of *PermaSense* is based on *Vizzly* [7], it can show a plot for some user-selected data, but there is no interaction between different data types, little flexibility to show time-lapse images, and no analysis method can be used to help understand the dataset. Another problem of the current interface is that when long-term data plot is requested, the peaks will be smoothed since the data is aggregated before plotting. This causes the loss of data features, and the shape of the plot is distorted.

With the consideration of research requirements, an upgraded visualization tool is needed. That tool is supposed to have these following functions:

- Plotting different datasets dynamically and interactively in a relatively short time regardless of required time frames.
- Analyzing the seismic stream with several common methods.
- Showing the time-lapse images more flexible.

Most data visualization tools in the market, like sisense [8], just plot and analyze the input data from statistic or business field. There is no customized tool for geophysical data which is able to show the datasets dynamically and do some analysis. In that case, such a visualization tool needs to be self-designed and constructed.

PermaSense offers different kinds of geophysical data [5], such as weather information, rock temperature, seismic stream, radiometer, GPS, images and so on. From this large variety of sensors, four typical measurements are considered in this project, i.e., weather information, rock temperature, seismic stream, and image. A Python tool [6] can be used to access these four datasets.

The plotting time scales with the amount of data, so it is impossible to get a long-term graph quickly with the use of original data, and sometimes it may cause the browser crashing. In that case, suitable downsample methods are necessary to plot these datasets without distorted visual effects. Chapter 2 introduces two different downsample methods used in this visualization tool, and how to compare the difference between images to measure the visual effects. As stated before, analysis of seismic stream should be included in the interface, so two simple methods are briefly explained in Chapter 2 as well.

Following the requirements for the visualization tool, five parts are designed to construct this interface with the use of *Dash* [9], which is a Python framework for building analytical web applications. More details regarding overview design, data preparation, choice of parameters, interaction between different data types and functions for different *Dash* components will be discussed in Chapter 3.

Chapter 4 shows speed performance and visual effects of weather information plot, seismic stream plot, and spectrogram plot respectively.

Chapter 5 discusses this visualization tool's achievements and weaknesses. Following this discussion, a conclusion and some potential improvements are put forward in Chapter 6.

Chapter 2

Methods

2.1 Downsample Methods

Due to the large capacity of datasets, it takes a long time to plot a series of long-term data, and sometimes it will cause the browser crashing. In order to speed up the plotting process, and avoid the situation which may crash the browser, suitable downsampling methods are used to reduce the number of points shown in one plot while guaranteeing a good visual effect. In this section, two downsampling methods are introduced, both of them separate the datasets into several buckets and try to find good representations for these buckets.

2.1.1 LTTB Downsample Method

Largest-Triangle-Three-Buckets algorithm [10] is used to downsample time series data. The principle behind this algorithm is first splitting the data into several buckets, the first bucket b_1 only contains the first data point p_1, the last bucket b_2 contains the last data point p_2 and the rest buckets contain an equal number of data points, the number of points is known as $DownsampleRate$. For each bucket except b_1 and b_2, one data point is selected to represent all points within this bucket. In order to select it, for each point p, the area of a triangle formed by p, the average point in the next bucket and p_2 is calculated, then the point corresponding to the largest area is chosen. The downsampled dataset consists of all selected data points, p_1, and p_2.

Unlike uniform downsample method, the LTTB downsample method can retain the fluctuations of time series data, so the local peaks will not disappear because of downsampling. This feature of the LTTB downsample method helps to preserve the local peaks of the observed time series data, so the user is able to know its minimum or maximum value even from a downsampled data frame.

A Python package [11] based on Numpy can be used to implement the LTTB algorithm.

2.1.2 Maximum Minimum Downsample Method

If a signal vibrates greatly within a short period, in each bucket, both the lowest and highest values are requested to avoid distorting the plot's shape. In that case, the LTTB algorithm is not suitable anymore, and Maximum Minimum Downsample Method is implemented.

The principle behind this algorithm is similar to a waveform plotting function [12]. First splitting the data points uniformly into several buckets, the number of data points in each bucket is known as *DownsampleRate*. Then for each bucket, the data points with the highest and lowest values are selected to represent all data points in this bucket. After processing all buckets, the downsampled dataset consists of all selected highest and lowest values.

The Maximum Minimum Downsample Method can preserve the vibration features of time series data, so it is quite suitable for downsampling time series data which may have significant change suddenly like the seismic stream.

2.2 Seismic Stream Analysis Methods

2.2.1 Fast Fourier Transform

As indicated in [13], Fourier analysis is quite useful to analyze periodic phenomena like vibrations and wave motion. Fourier analysis converts a signal from its time domain to a representation in the frequency domain. The Discrete Fourier Transform(DFT) is obtained by decomposing a discrete sequence of values into components of different frequencies. Fast Fourier Transform (FFT) algorithm is used to help calculate DFT efficiently. In this visualization tool, a function [14] from Numpy package is used to compute FFT for seismic stream online.

2.2.2 Spectrogram

With the use of Fourier transform, the seismic spectrum can be decomposed into constituent frequency components as discussed in Chapter 2.2.1. However, their temporal locations cannot be identified. Under that circumstance, time-frequency distributions (TFDs) can be used to map a one-dimensional signal into a two-dimensional function of time and frequency, and describe how the spectral content of the signal changes with time [15]. One of TFDs is spectrogram (SP), which is well suited to identify the dominant frequencies contained in the seismic signals [15]. It is used to analyze the seismic stream in this project.

As shown in Figure 2.1, a spectrogram has 3 dimensions, one of the two geometric dimensions represents time, the other geometric axis is frequency, and the third dimension indicates the intensity of a particular frequency at a specific time which is represented by the color of each point in the graph. In this visualization tool, a function [16] from Scipy package is used to compute spectrogram for the seismic stream.

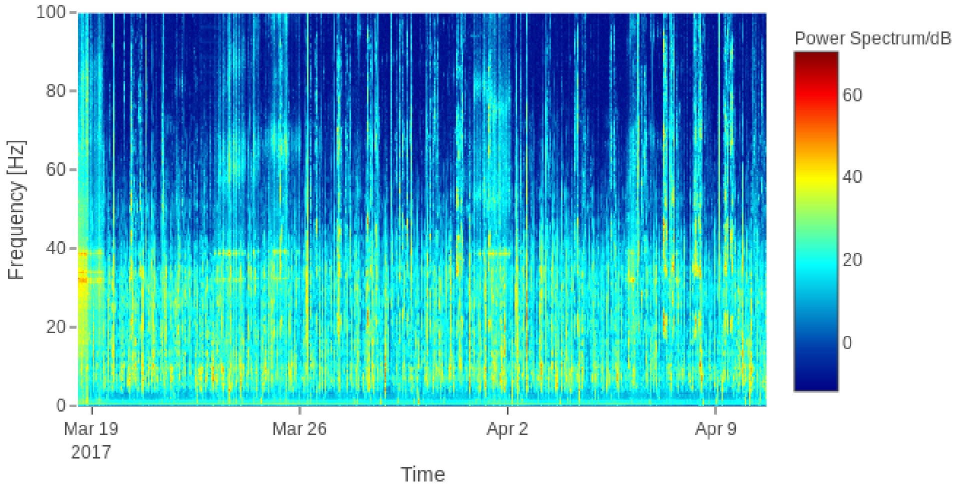

Figure 2.1: An example of spectrogram

2.3 Difference Between Images

The quality of downsample result can be quantified by the difference between the original plot and the downsampled ones. In this section, two common metrics are introduced, which can help analyze the difference between two images from the numeric side.

2.3.1 Mean Square Error

Mean Square Error (MSE) is a traditional and simple method for measuring the energy of difference between original and test images. In order to calculate MSE, two grey scale images are compared pixel by pixel to calculate the square of difference between error of original and test images [17], as shown in Equation 2.1. The square of difference can be replaced by the number of different pixels or the absolute value of difference, i.e, Zero norm and Manhattan norm for the subtraction of two images.

$$MSE = \frac{1}{MN}\sum_{i=1}^{N}\sum_{j=1}^{M}(x_{i,j} - y_{i,j})^2 \tag{2.1}$$

Before calculating the MSE, the original and test images need to be converted into grey-scale and normalized.

2.3.2 Structural Similarity Index

The Structural SIMilarity (SSIM) index is one of the methods to measure the similarity between two images, it can be viewed as a quality measurement provided one of the images being compared, while the other image is regarded as of perfect quality, more mathematics details can be found in [18]. Unlike MSE, SSIM index is independent of visibility conditions, and it extracts structural information from images [17].

Chapter 3

Results

The current visualization tool consists of 5 parts, including plotting weather information or rock temperature, plotting seismic stream, calculating FFT of seismic stream, showing spectrogram of seismic stream, and showing time-lapse images. Figure 3.1 shows how this visualization tool looks like, details about each part will be discussed in this chapter.

3.1 Structure

Figure 3.1 shows the internal structure of this visualization tool. After the datasets are collected, some preprocessed files are generated, then according to the time frame selected by the user, this visualization tool dynamically chooses different files. The change of the set time frame in one graph may cause the time frame updating for some other graphs. In this flowchart, blue lines indicate the source of datasets and no extra analysis is used, the green lines mean the analysis methods for the seismic stream, the purple lines represent the dynamic interaction among different data files which belong to the same data type, and the red lines show how the time frames update once the time frame changes in one of these graphs.

3.2 Weather Information

The weather information got by *PermaSense* contains 10 different types of data, including minimum wind direction, average wind direction, maximum wind direction, minimum wind speed, average wind speed, maximum wind speed, air temperature, internal temperature, relative humidity, and air pressure. Because of redundant items, average wind direction is used to present wind direction information, and average wind speed is used to present wind speed information. So there are 6 different types of weather information shown in the visualization interface.

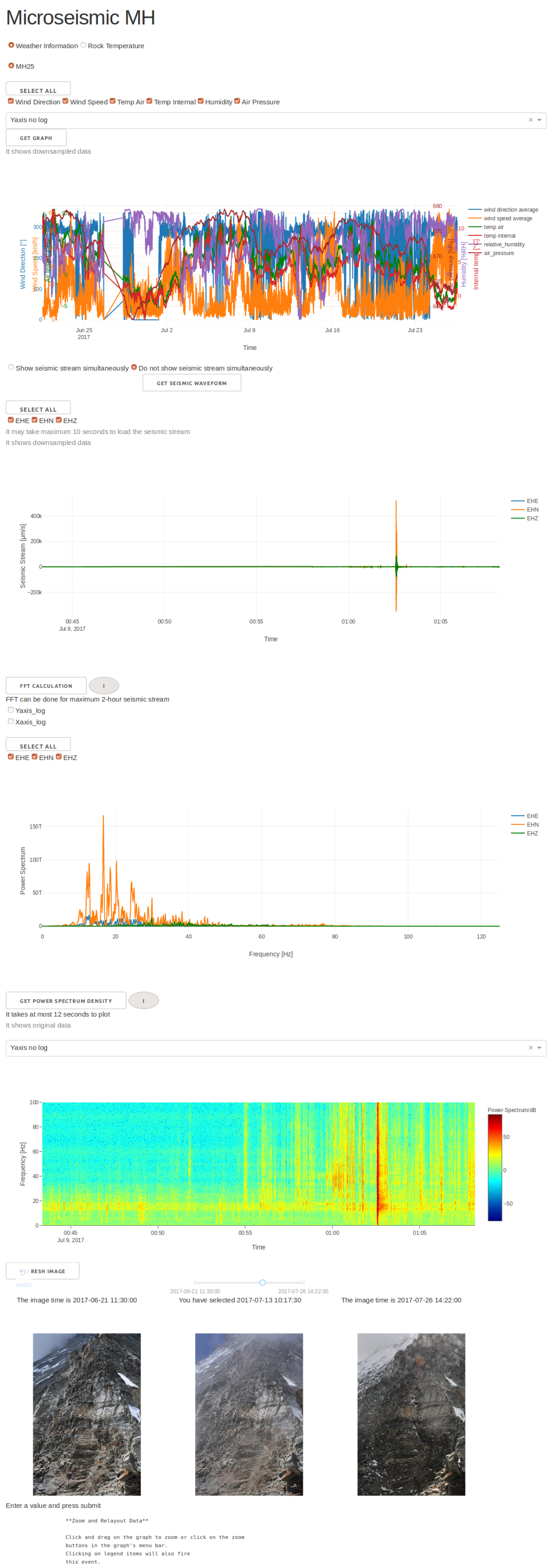

Figure 3.1: The visualisation tool

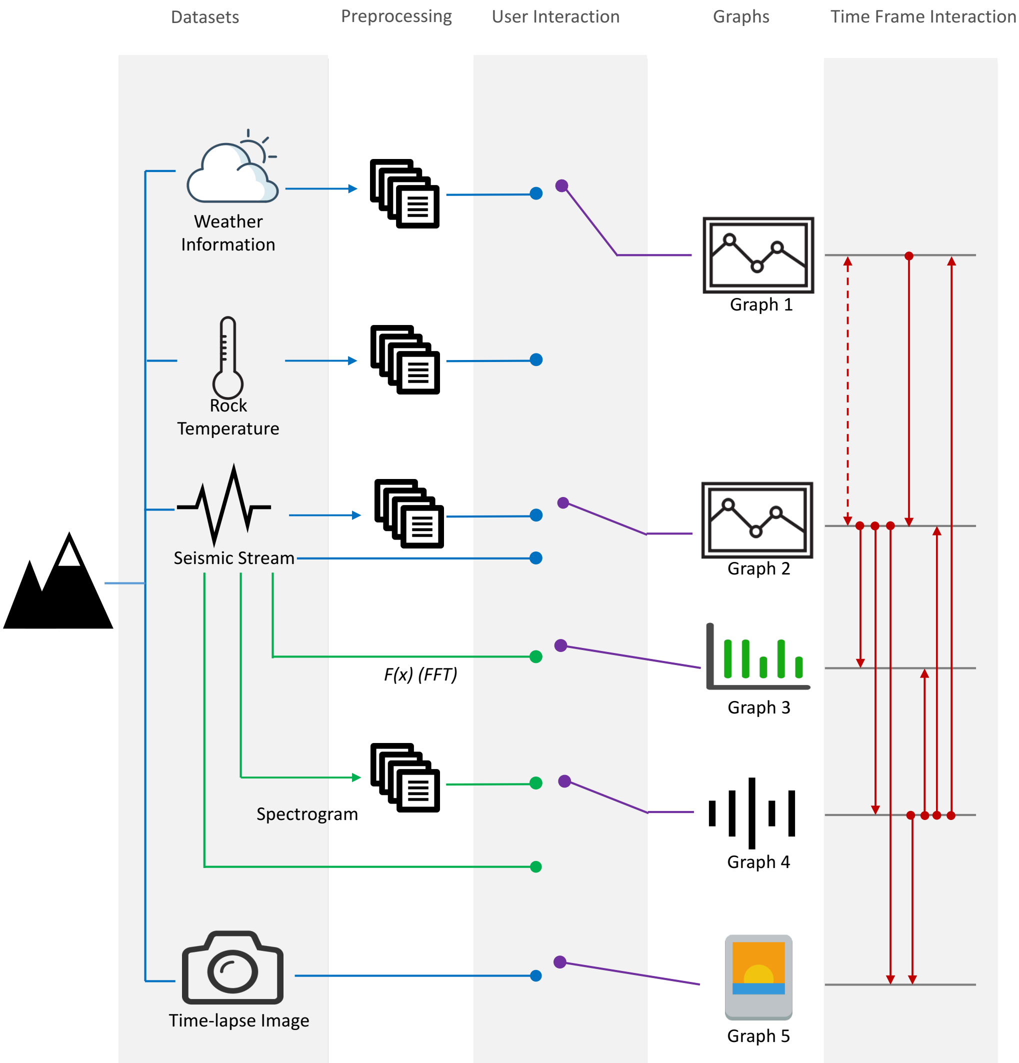

Figure 3.2: The visualisation tool's flowchart

Before processing the weather information, it can be noticed that the weather information dataset contains some unrealistic data possibly caused by occasional sensor issues, for example, the recorded temperature may be below $-30°C$. In order to eliminate these data points, thresholds are set to delete data points where air temperature is below $-50°C$, internal temperature is below $-30°C$ or air pressure is less than 500 Pa. Besides, sometimes the dataset may have some missing data points, and it will cause the graph to plot discrete points instead of continued lines. So, the NaN values are dropped from the data stream.

Because of weather information's characteristics, it can be known that this type of data doesn't fluctuate rapidly within a short time frame. Then considering the advantages of LTTB algorithm illustrated in Chapter 2.1.1, it is a good option to downsample weather information. After that, the plotting process can be speeded up since the number of plotted points decreases. If the downsampling process is completed online, extra time is needed to load the original data and do the calculation. In that case, the downsampling process is completed offline and downsampled datasets corresponding to different *DownsampleRate* are stored in the local disk.

In order to achieve similar speed performance for different time frames, different *DownsampleRate*s are needed to be chosen. In order to get a balance between fast plotting speed and good visual representation, after several experiments with original data, the expected maximum plotting time θ is set to be 1 second. The time frames plan to be separated into several ranges, in each range, the visual effect is measured at the minimum time frame while the average plotting time is tested at the maximum time frame.

To measure the visual effects of the downsampled data, 12 different times frames whose time frames length is 20 days are selected, then plots are got with the use of original or downsampled data with different *DownsampleRate*. Metrics introduced in Chapter 2.3 are used to see if there exists one metric can reflect the downsampled effect accurately and stable. Figure 3.3 shows the average Manhattan norm between the original plot and the downsampled one, it can be seen that in general smaller *DownsampleRate* causes smaller error, but it is not always true with the consideration of standard deviation.

MSE, SSIM index, and average zero norm are tested following the same procedure, it turns out that all of them cannot guarantee a smaller *DownsampleRate* will generate a smaller error. Under that circumstance, all of these metrics are not good enough to quantify the downsampled error. However, since the visual effect is about the human perception of the visual system, the easiest solution is to let the user subjectively determine whether a good visual effect is achieved or not.

As a smaller *DownsampleRate* can achieve better visual effects, it is expected that a smaller one can be used for a longer time frame. In other words, for each *DownsampleRate*, the maximum time frame whose average plotting time is less

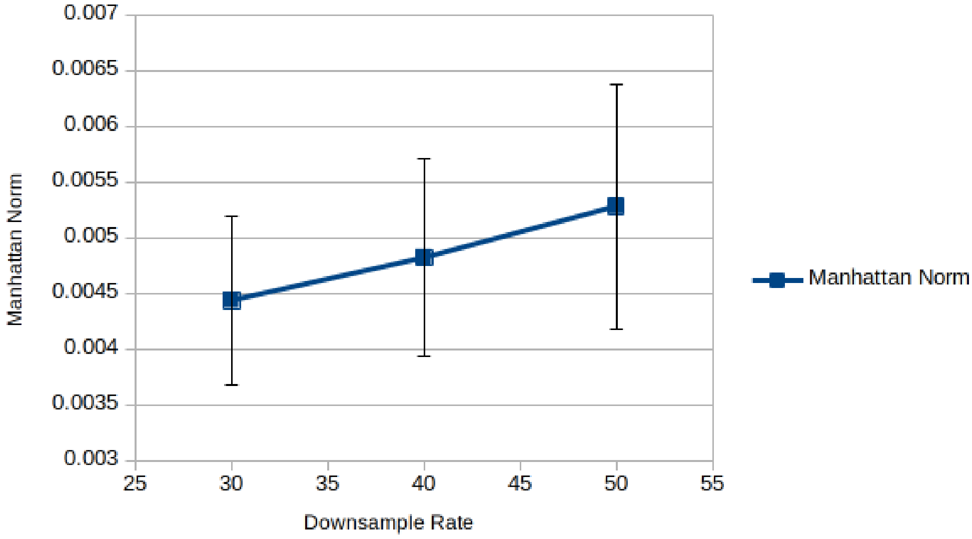

Figure 3.3: The Manhattan norm between plots using original data and down-sampled data

than the fixed threshold θ needs to be found. The average plotting time for a certain time frame is calculated by 10 repeated experiments.

This selection process starts with the original data, and before using a new *DownsampleRate*, the visual effect is checked at the last maximum time frame, if the visual effect isn't ideal, smaller *DownsampleRate* will be attempted. For example, when 50 is used as the *DownsampleRate*, first the visual effect is checked with 90-day weather data as shown in Figure 3.4, 90-day is the maximum time frame to use *DownsampleRate* 20. Since it is difficult to observe 6 types of weather information, only air temperature and air pressure are plotted there, and other types of weather information are observed as well. Then, the plotting time for several time frames is tested as shown in Figure 3.5 to get the upper bound. It can be known that if the maximum time frame whose *DownsampleRate* is 50 is 210 days or more, the average plotting time exceeds θ. In that case, only time frames less than or equal to 209 days use 50 as its *DownsampleRate*. Following this procedure, Table 3.1 is got to reflect the corresponding relationship between *DownsampleRate* and time frames.

The weather information part in the current interface is like shown in Figure 3.6, it corresponds to the graph1 in Figure 3.1. The user can select to show weather information or rock temperature in that graph, they can also choose to show data got by different sensors installed at different locations if there are multiple ones can be used like shown in Figure 3.7. There are 6 different types of weather information can be shown, the user can also choose to plot only some

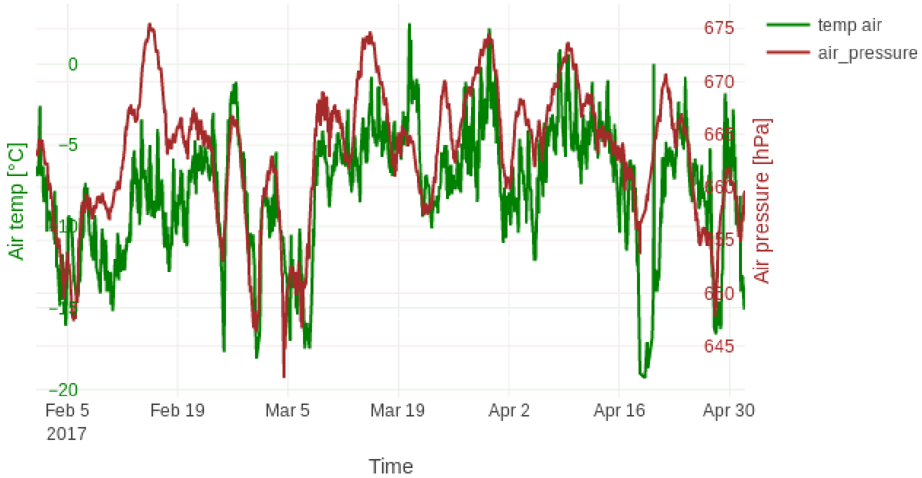

Figure 3.4: 90-day partial weather information with *DownsampleRate* 50

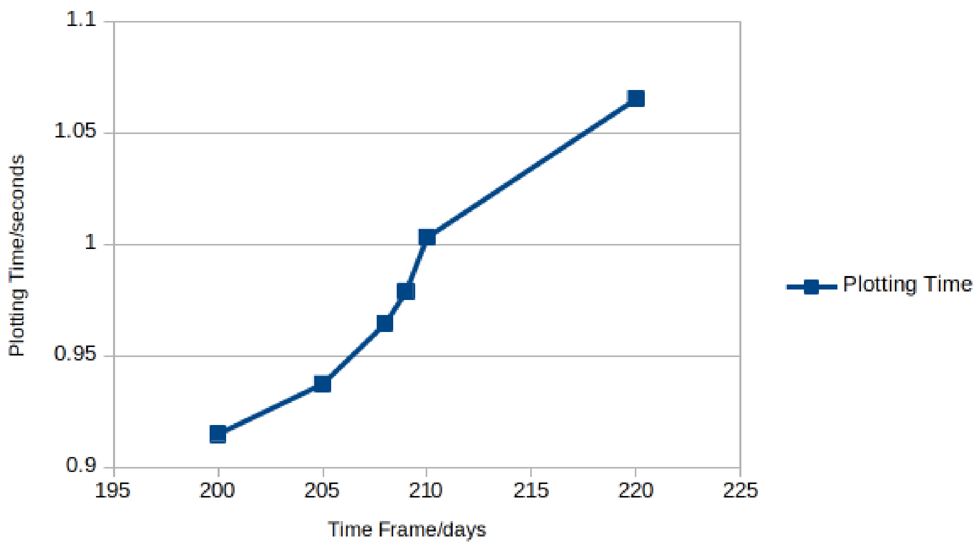

Figure 3.5: Average plotting time for *DownsampleRate* 50

Time Frame /days	*DownsampleRate*
<4	1
4-18	5
18-42	10
42-90	20
90-209	50
209-400	100
$\geq$ 400	200

Table 3.1: The *DownsampleRate* for weather information

of them. Sometimes it is more convenient to observe data information under the logarithm axis, the user is free to use logarithm for Y-axis or not. If the logarithm axis is used, a warning message will be shown in this interface if there is any negative value, since in that case, the plot may look weird. Besides, in order to help the user have an idea if the graph plots the original data or the downsampled one, an explanation is included.

3.3 Rock Temperature

The rock temperature got by *PermaSense* contains 6 different types of data, which are got 5 cm, 10 cm, 20 cm, 30 cm, 50 cm, and 100 cm underground respectively. Unlike the weather information, rock temperature isn't found containing any unrealistic data, so thresholds which are used to eliminate them aren't necessary anymore. However, there are still some missing data points in the dataset, like the treatment for weather information, the NaN values are dropped from the data stream.

Similar to weather information, rock temperature can be downsampled by the LTTB algorithm as well. Because rock temperature data has the same frequency as weather information data, *DownsampleRate* in Table 3.1 is directly used for preprocessing procedures. This visualization tool shifts between different downsampled rock temperature depending on the selected time frame.

The rock temperature part of the current visualization tool is like shown in Figure 3.7, it shares the same graph with weather information, and the selected time works on both of them. The arrangements are quite similar to weather information component in Chapter 3.2, except there are two sensors collecting rock temperature from different positions, and users can indicate which one to be used.

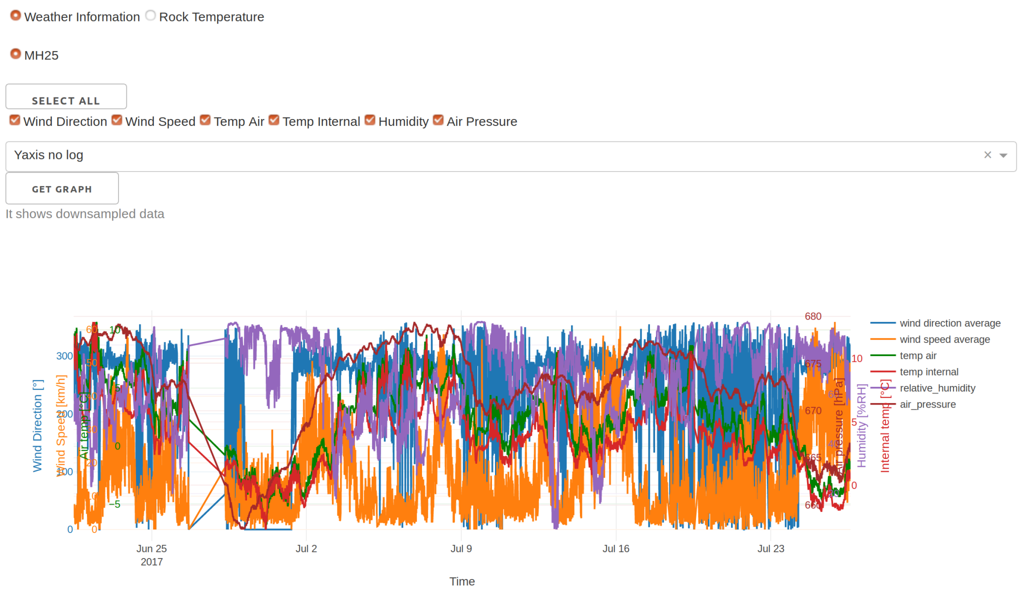

Figure 3.6: The weather information part of the visualization tool

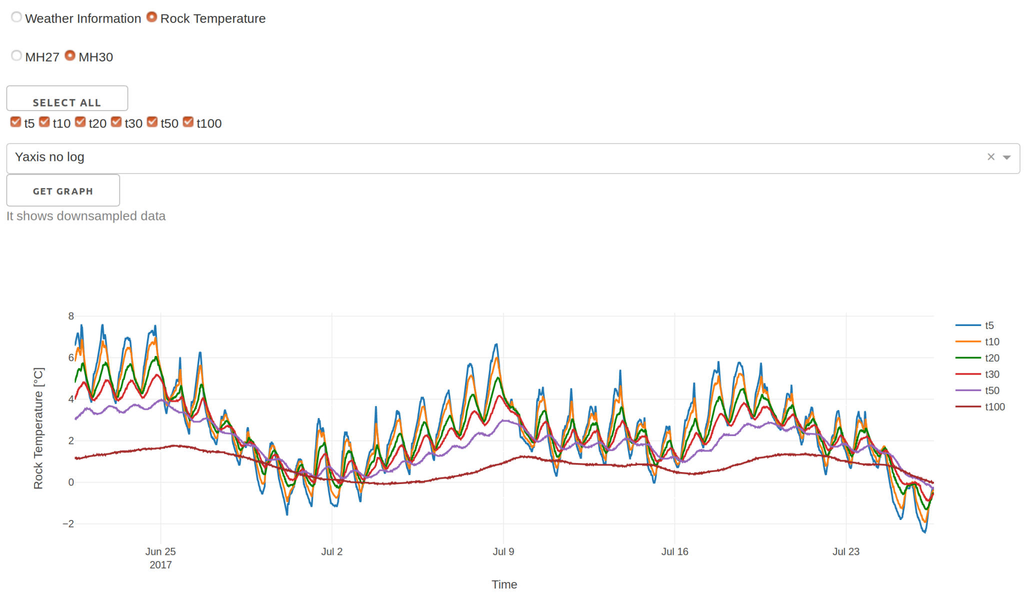

Figure 3.7: The rock temperature part of the visualization tool

3.4 Seismic Stream

3.4.1 Plotting Seismic Stream

PermaSense collects seismic stream 1000 times per second, it indicates that a large amount of data points needs to be plotted in one long-term graph and the user has to wait a long time. For example, it takes approximately 50 seconds to plot 1-hour seismic stream with the use of original data points. Besides, it will also cause the browser crashing if the user wants to plot seismic stream for more than 2 hours. Under that circumstance, suitable downsample method is needed to enable plotting seismic stream within a long time frame, $Maximum$ $Minimum$ downsample method can be used as discussed in Chapter 2.1.2.

Like what we do to weather information, in order to achieve similar speed performance for different time frames, different *DownsampleRate*s are needed. In order to get a balance between potential fastest plotting speed and good visual representation, the expected maximum plotting time θ is set to be 5 seconds.

For the seismic stream, the quality of downsampled data cannot be quantified not only because metrics listed in Chapter 2.3 don't achieve stable performance as illustrated in Chapter 3.2, but also because it is impossible to get a long-term seismic stream plot with the use of original data, which will cause the browser crashing. In that case, the visual effect is measured by the subjective feeling as well.

The time frames are separated into several ranges with a similar method introduced in Chapter 3.2. Through 10 repeated experiments with different time frames, the corresponding relation between *DownsampleRate*s and range of time frames is like shown in Table 3.2. For seismic stream whose time frame is less than 4 hours the downsample processing is done online, otherwise, the downsample processing is completed offline. During the preprocessing stage, two downsampled seismic stream files are stored with *DownsampleRate* 5000 and 36000 respectively.

Time Frame	*DownsampleRate*	Online/Offline
<5 minutes	1	Online
5-10 minutes	4	Online
10-30 minutes	12	Online
30 minutes-4 hours	450	Online
4 hours-10 days	5000	Offline
10-83 days	36000	Offline
$\geq$ 83 days	360000	Offline

Table 3.2: The *DownsampleRate* for seismic stream

When seismic stream needs to be plotted, the visualization tool decides whether the downsampled data points can be loaded directly from preprocessed files or the downsampled procedure needs to be completed online.

The seismic stream part of the current visualization tool is like shown in Figure 3.8. The time frame for the seismic stream plot is the one indicated by the user in weather information or rock temperature plot, the mutual influence can be either unilateral or bilateral like shown by the red solid line and dashed line between graph1 and graph2 in Figure 3.1. If the user chooses the "Do not show seismic stream simultaneously" radio item, the time frame in the weather information or rock temperature plot is used as an initial time frame for the seismic stream plot, but the change of time frame in the seismic stream plot cannot be sent back to the weather information or rock temperature plot. However, if the "Show seismic stream simultaneously" item is chosen, the change of the selected time frame in the weather information or rock temperature plot causes one update for both plots and vice versa.

The seismic stream has three different data components which correspond to seismic signals at three orthogonal directions, like weather information and rock temperature, users can choose to show only some of them. There are two prompt messages in this part, one indicates that the loading process may take several seconds, another shows the plot uses original data or downsampled data. Besides, in order to avoid the user sending repeated requests when they don't know if the visualization tool is working, the whole interface becomes grey when there is any running task.

3.4.2 Analysis Of Seismic Stream – FFT

As discussed in Chapter 2.2, Fast Fourier Transform(FFT) and spectrogram can be used to analyze seismic stream from the frequency domain.

Since we care more about the frequency features caused by instant changes of the seismic stream, there is no need to do FFT for a long time frame. The maximum time frame is set to be 2 hours.

The FFT part of the current visualization tool is like shown in Figure 3.10. One information button is added to explain the aim of computing FFT and show relative parameters. Similar to weather information, users can choose to use logarithm for X-axis and Y-axis in order to get a better understanding of the seismic stream in frequency domain like shown in Figure 3.11. The Seismic stream contains 3 components as explained in Chapter 3.4.1, users are free to analyze only some of them in the frequency domain, for example, Figure 3.11 plots the FFT result of the seismic stream in 2 orthogonal directions.

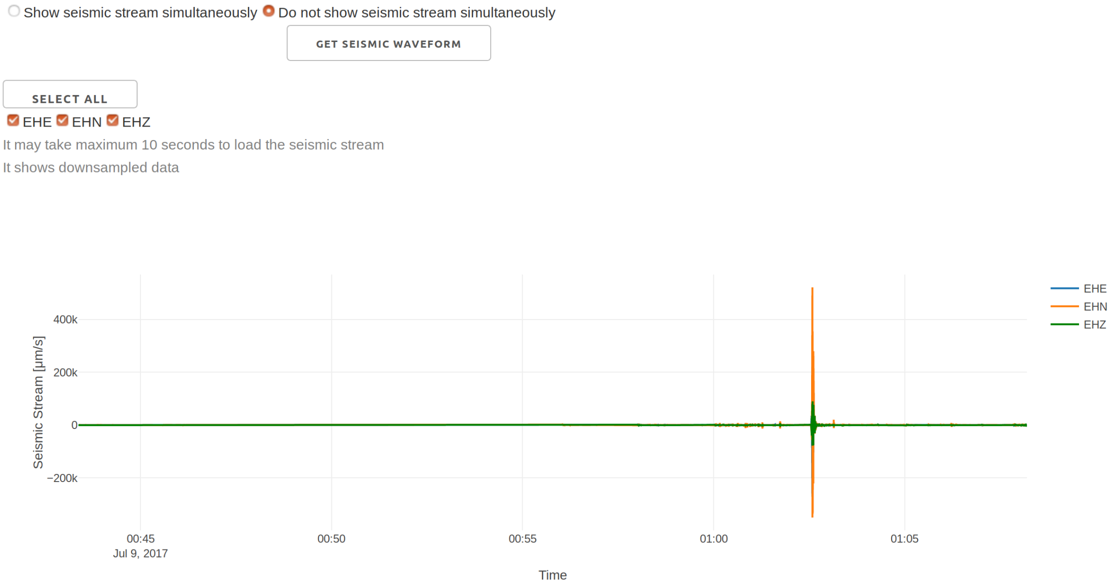

Figure 3.8: The seismic stream part of the visualization tool

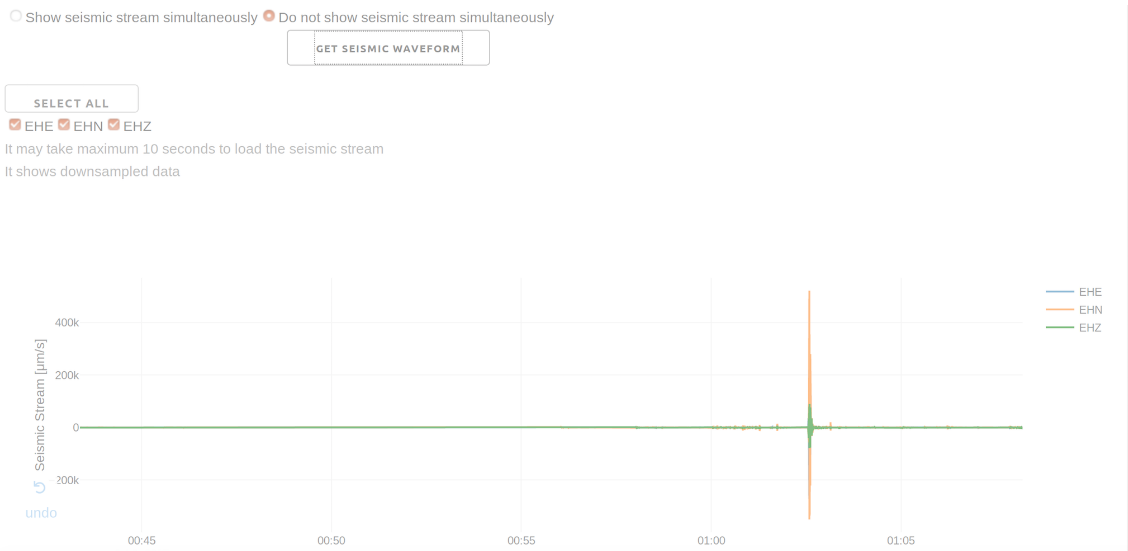

Figure 3.9: The visualization tool when there is any running task

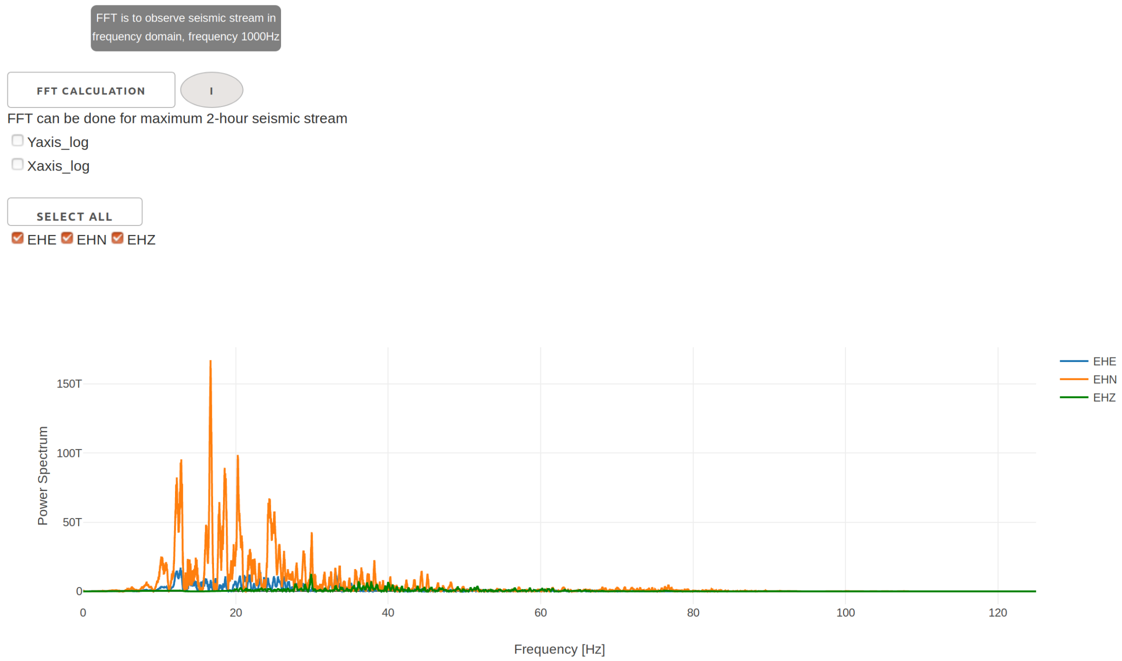

Figure 3.10: The FFT part of the visualization tool

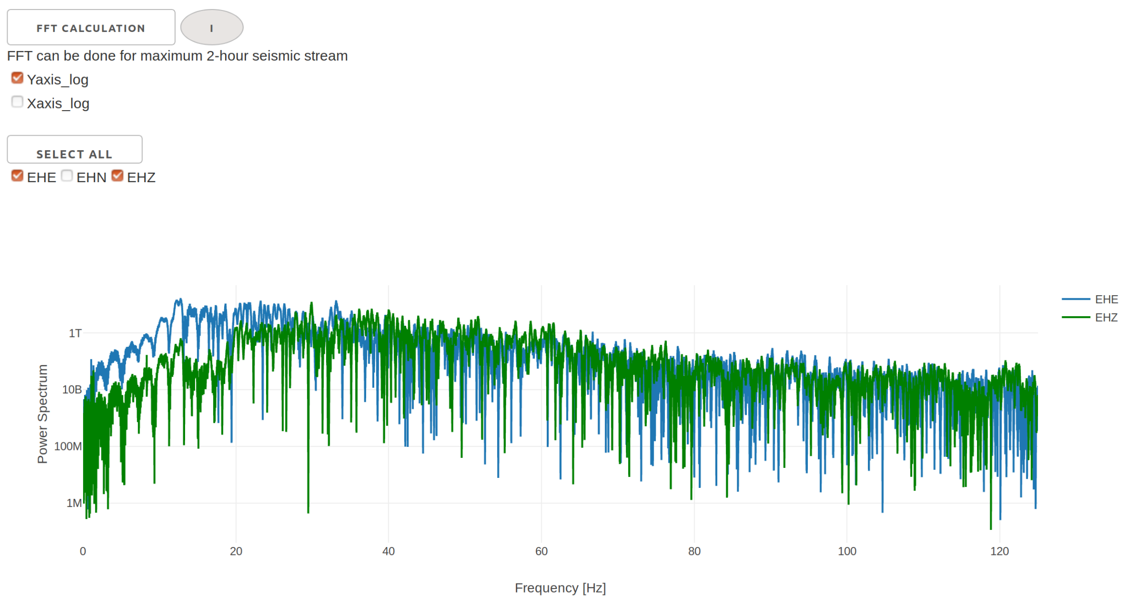

Figure 3.11: An example of FFT with two components using logarithm Y-axis

3.4.3 Analysis Of Seismic Stream – Spectrogram

In order to analyze the seismic stream from power spectrum aspect, the spectrogram is used as discussed in Chapter 2.2.2. Considering the sensitivity of relative sensors, only frequencies which are less than 100 Hz can be measured accurately, so only this part is taken into consideration when plotting the spectrogram.

Because of the high sampling frequency, it takes a long time to get a spectrogram, for example, it will need 10 seconds to calculate and plot a spectrogram for a 3-hour seismic stream. It is also impossible to show a spectrogram for a month's seismic stream since it will cause the browser crashing. In that case, a suitable downsample method is required.

From Chapter 2.2.2, it can be known that the spectrogram's color bar indicates the intensity of power spectrum at a particular frequency and a specific time. When downsampling the spectrum, it needs to be done from the time axis, i.e, the X-axis of the spectrogram. The time axis can be uniformly separated into several buckets, for each bucket, a new column is needed to represent all data points in this bucket. In order to get this column, the easiest method is to select one column from each bucket, or a new column can be generated by combining the maximum value at each frequency. With the use of the first method, the selection of each column is a bit random, and intensity features will be lost during this process. In contrast, if a new column is generated as discussed before, the information about power spectrum density can be kept and the user is able to get such knowledge even through a downsampled spectrogram. In that case, the second downsample method is preferred, its procedure is explained in Figure 3.12.

Unlike what is done to the seismic stream in Chapter 3.4.1, it is better to let each column in the downsampled spectrogram have some actual meaning, in order to help the user understand how much data is represented by one column. To achieve that, the original spectrogram is downsampled by letting one column represents the dataset within 1 minute, 10 minutes, 1 hour, or 1 day. The correspondence between different time frames and the meaning of one column in the downsampled spectrogram is listed in Table 3.3.

Time Frame	One Column Is Data Within
<3 hours	Original
3 hours - 3days	1 minute
3 - 30 days	10 minutes
30 - 300 days	1 hour
$\geq$ 300 days	1 day

Table 3.3: The correspondence between time frames and how much data is represented by one column in the downsampled spectrogram

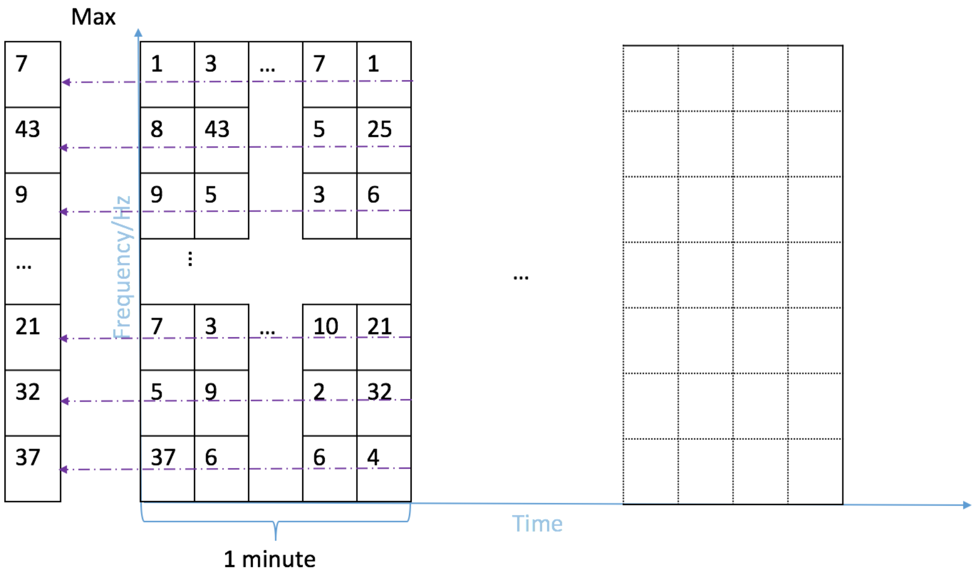

Figure 3.12: A sketch map for spectrogram downsampling process

The spectrogram part of the visualization tool is like shown in Figure 3.13. Like shown in Figure 3.1, the spectrogram's initial time frame is the last modified one in graph1 and graph2. There is an information button to indicate the aim of spectrogram and relative parameters, and a hint to let the user have an idea about the meaning of one column in the downsampled spectrogram. Besides, the user can show the frequency with logarithm Y-axis.

3.5 Time-lapse Image

PermaSense includes some sensors which take pictures for the Swiss Alps at the specific locations. These time-lapse images can give users an intuition about how the mountain looks like at a particular time. The time-lapse image part of the current visualization tool is like shown in Figure 3.14.

The left picture shows how the mountain looks like at the beginning moment of the selected time frame while the right one is a picture taken at the end moment. The moment corresponding to the middle photo can be shifted by users through a range slider, which contains 40 discrete time moments uniformly distributed between the start point and end point of the time frame.

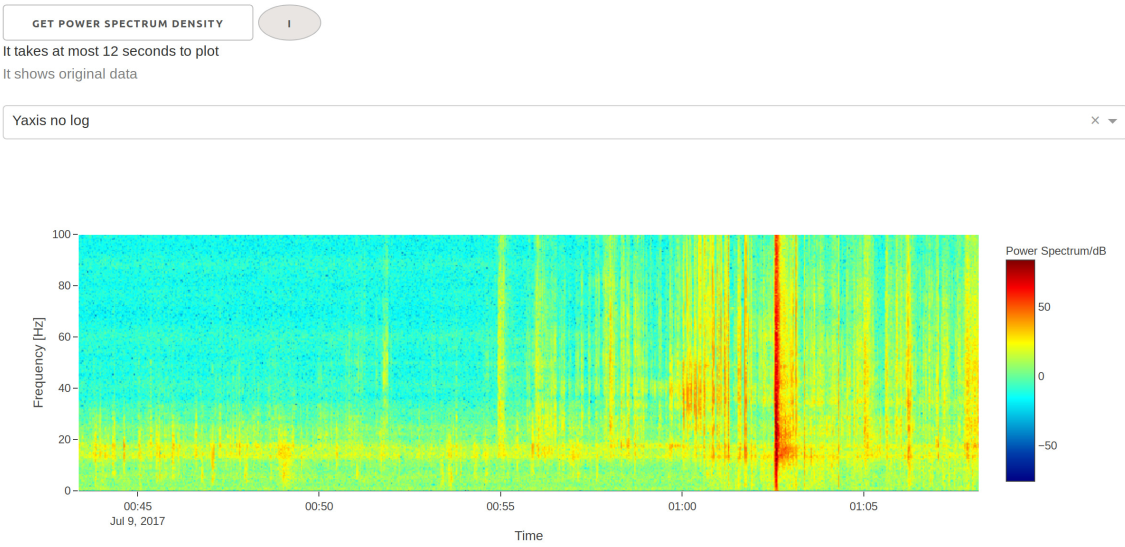

Figure 3.13: The spectrogram part of the visualization tool

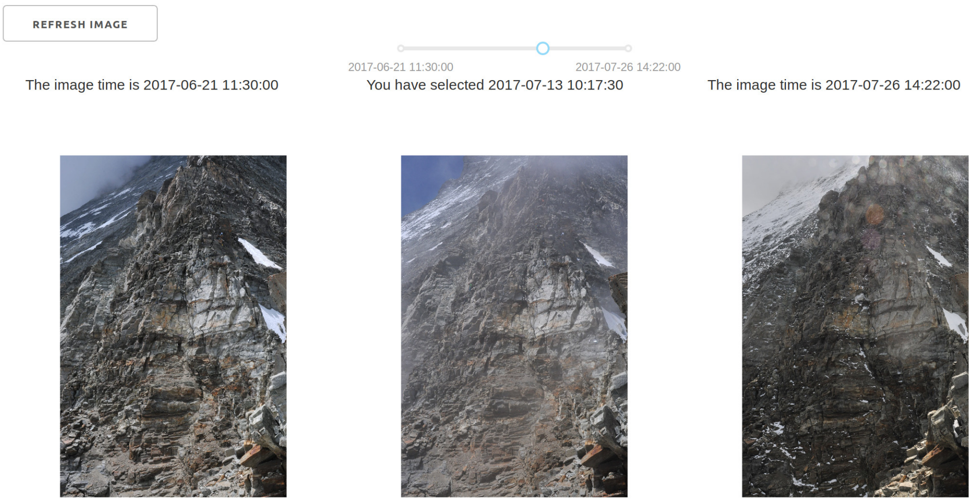

Figure 3.14: The time-lapse image part of the visualization tool

Chapter 4

Evaluation

4.1 Weather Information

Like discussed in Chapter 3.2, the maximum plotting time θ for weather information is expected to be less than 1 second. In order to evaluate that, several tests are performed for different ranges of time frames, the result is shown in Figure 4.1. From that, it can be observed that the expectation of plotting time can be met with the use of different *DownsampleRate* for different time frames.

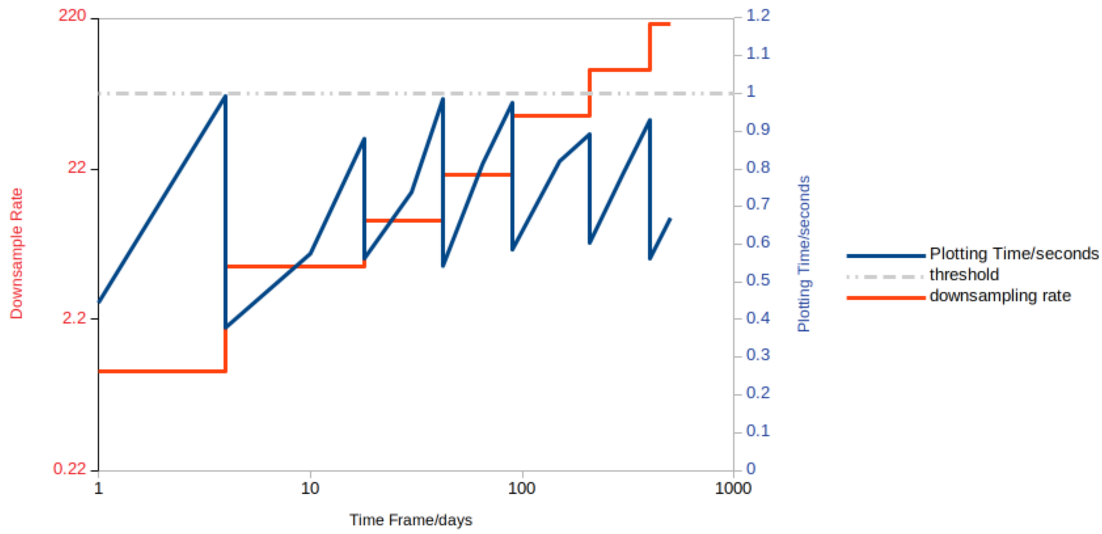

Figure 4.1: The plotting time for weather information

Besides the plotting time, we also care about the visual effect, since the large *DownsampleRate* may cause a distorted signal. However, it is hard to quantify the visual effect as discussed in Chapter 3.2, so the visual effect is tested by the developer at the minimum time frame of each time frame range.

Figure 4.2 shows a plot of two types of weather information with time frame 18 days which is the minimum time frame for range 18-42 days, as shown in Table 3.1, the *DownsampleRate* is 10. Figure 4.3 shows a plot of two types of weather information with time frame 209 days which is the minimum time frame for range 209-400 days, as shown in Table 3.1, the *DownsampleRate* is 100. In order to make it convenient to observe, only air temperature and air pressure are plotted, but the downsampled effect is checked with the use of other types of weather information as well. From these, it can be verified that a balance between a short plotting time and good visual representation has been achieved.

Figure 4.2: The air temperature and air pressure within 18 days

4.2 Seismic Stream

Following the discussion in Chapter 3.4, the maximum plotting time θ for the seismic stream is expected to be at most 5 seconds. In order to check that, several tests are completed for each range of time frames, the result is shown in Figure 4.4. From that, it can be seen that in most cases the plotting time doesn't exceed 5 seconds, but there are some exceptions. However, in general, it can be said that the plotting time of the seismic stream is less than 5.5 seconds regardless of time frames.

Similar to weather information, we also care about the visual effect, which

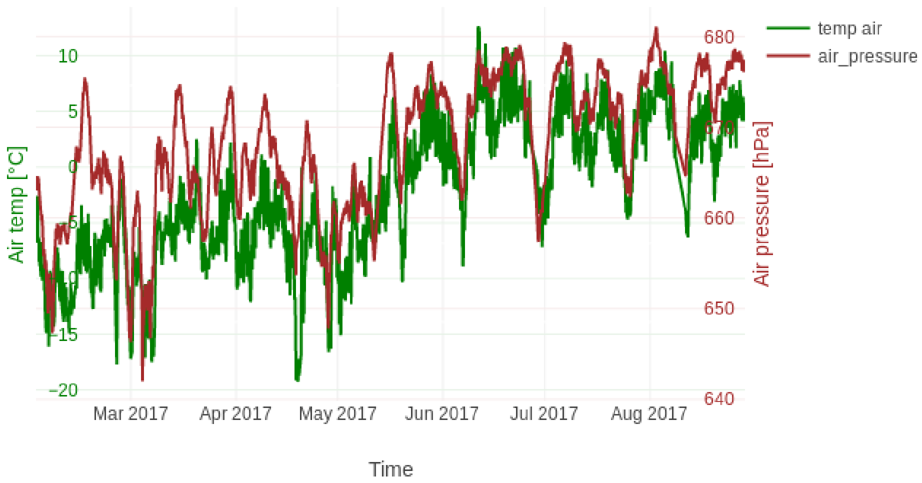

Figure 4.3: The air temperature and air pressure within 209 days

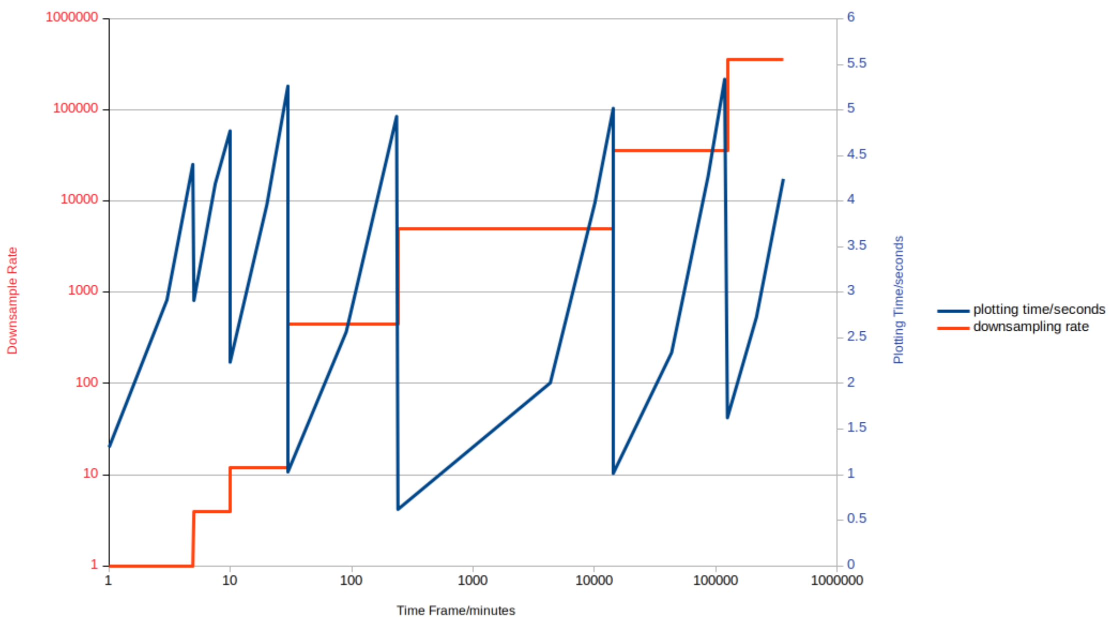

Figure 4.4: The plotting time for seismic stream

is hard to be quantified so measured by subjective feelings as well. The visual effect of the downsampled seismic stream is tested at the minimum time frame for each range similar to the evaluation for weather information.

Figure 4.5 shows a plot of the seismic stream with time frame 30 minutes which is the minimum time frame for range 30 minutes-4 hours, as shown in Table 3.2, the *DownsampleRate* is 450. Figure 4.6 shows a plot of the 10-day seismic stream, the time length is the minimum time frame for range 10-83 days, as shown in Table 3.2, the *DownsampleRate* is 36000. From these, it can be verified that a tradeoff between relatively short plotting time and good visual representation can be achieved.

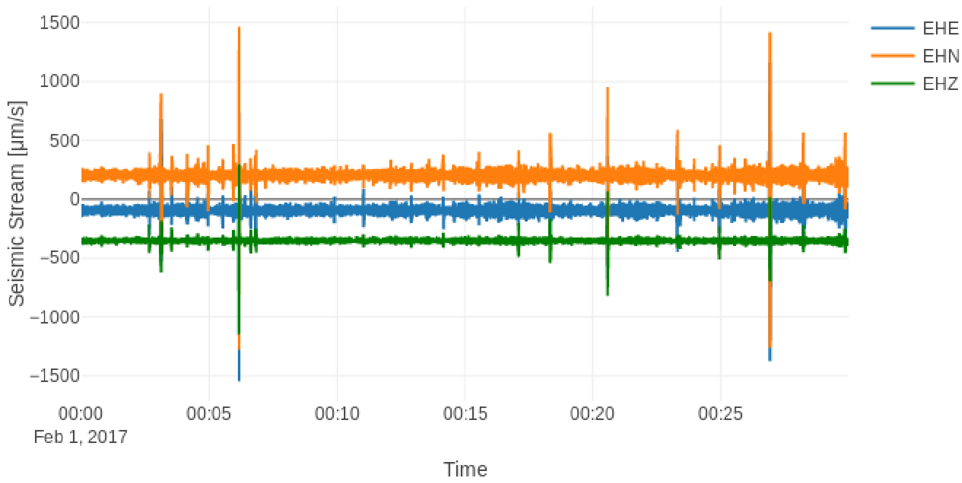

Figure 4.5: The seismic stream within 30 minutes

4.3 Spectrogram

As discussed in Chapter 3.4.3, downsampling method is used to speed up the plotting process of spectrogram and make it possible to show spectrogram for long-term data. In order to test the speed, experiments are completed for different time frames, the result is shown in Figure 4.7. From this graph, it can be known that a spectrogram can be plotted with at most 10 seconds regardless of its time frame as expected.

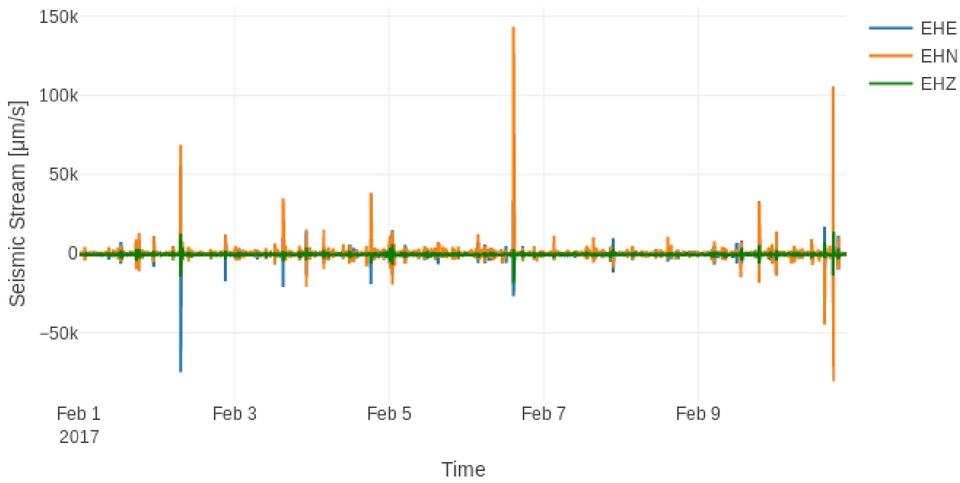

Figure 4.6: The seismic stream within 10 days

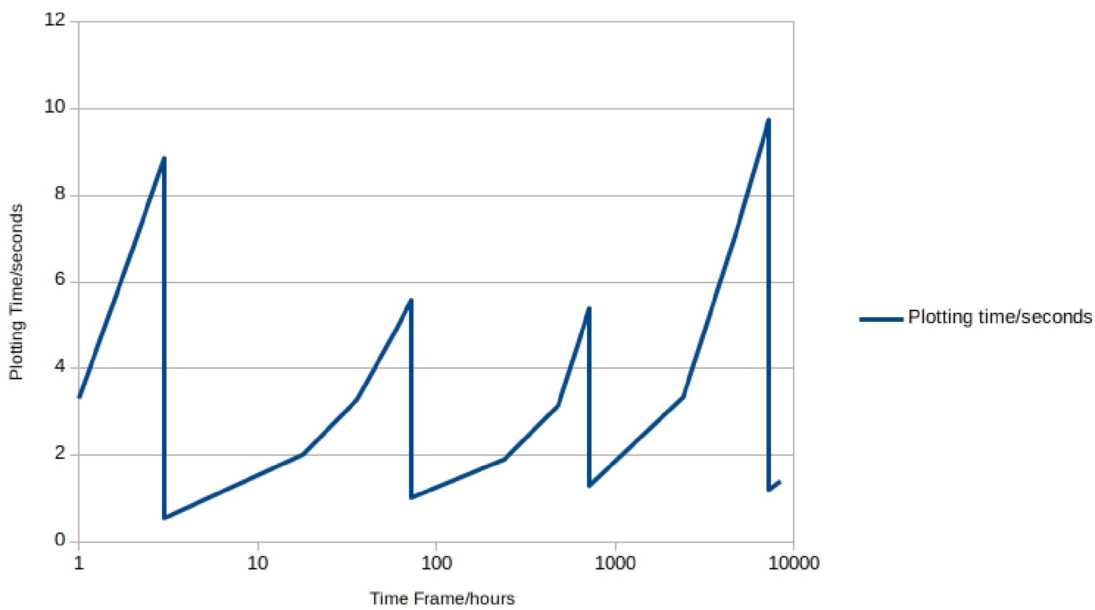

Figure 4.7: The plotting time for spectrogram

There are two figures showing the visual effect of the downsampled spectrogram. Figure 4.8 shows a spectrogram with time frame 3 days which is the minimum time frame for range 3-30 days, from Table 3.3, it can be known that each column represents 10-minute original spectrogram. Figure 4.9 is a 300-day spectrogram, its time length is the minimum time frame for range more than 300 days, from Table 3.3, it can be known that each column represents 1-day original spectrogram. From these, it can be known that the downsampling process is able to achieve an acceptable visual effect as requested.

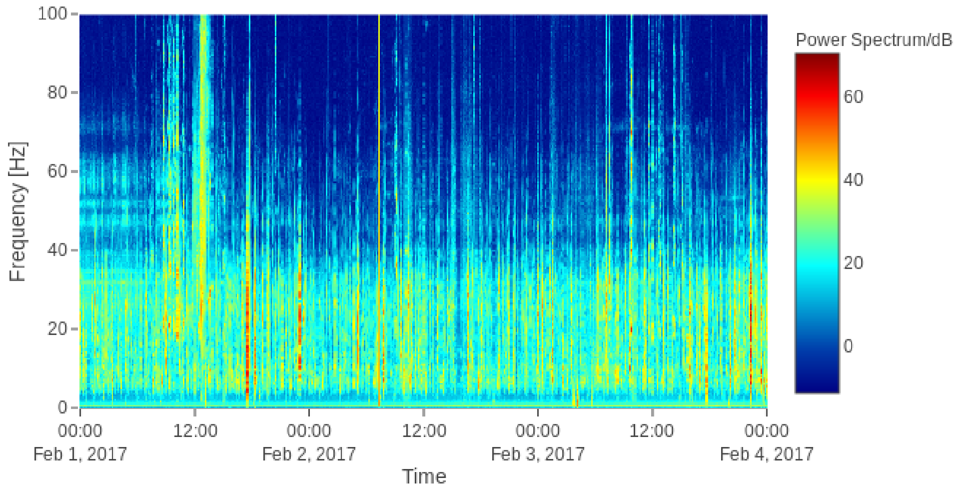

Figure 4.8: The spectrogram within 3 days

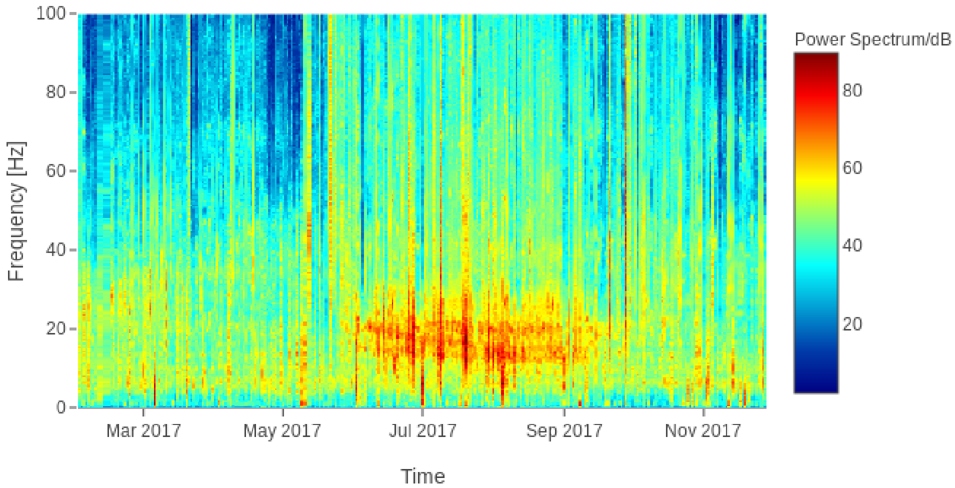

Figure 4.9: The spectrogram within 300 days

Chapter 5

Discussion

Following the requirements listed in Chapter 1, a dynamic visualization tool is created. The main difficulty through the project is caused by the large capacity of datasets and their intricate relationships. In order to overcome these challenges, suitable downsample and analysis methods are selected, and interaction between different graphs are implemented.

With the use of downsampling methods online or offline, the plotting processes can be speeded up regardless of data types and time frames. It is also verified that the user only needs to wait at most several seconds to get a requested plot. By analyzing the seismic stream, its features in the frequency domain can be understood better, while the user is able to get an intuition about its property in the time-frequency domain. Through the interaction among different graphs, the user can easily observe the different datasets' features for the selected time frame instead of adjusting plots repeatedly.

As discussed in Chapter 3, the operation of this visualization tool relies on some preprocessed files, for example, downsampled spectrogram files to speed up the plotting process. These files can be generated by some Python scripts, which need to be executed regularly by the admin in order to add the newest datasets.

Currently, the visual effects of plots with the use of the downsampled dataset can only be determined by subjective feelings. It is hard to quantify the downsample quality because of two main reasons. Firstly, it is difficult or impossible to get a plot with long-term data considering its capacity. If that plot cannot be accessed, the difference between plots with the use of original data and plots using downsampled data cannot be computed. Secondly, no common metrics can make sure smaller error is expected for smaller *DownsampleRate*, which contracts the aim of downsampling.

PermaSense contains hundreds of sensors, which can be simply classified into several data types. As illustrated in Chapter 1, this visualization tool is created with the use of four data types. This tool can be extended as long as using the similar processing method as used in this type of data. However, it means that modifications are needed to both arrangement of the interface and the internal programs.

For now, two analysis methods are used to analyze seismic stream from frequency domain and time-frequency domain respectively. However, the user may want to analyze the dataset from other fields. But, the current interface doesn't offer a port for the use of self-designed or self-chosen analysis methods.

Chapter 6

Conclusion and Future Work

In this project, a dynamic visualization tool is created for *PermaSense*. In order to satisfy the requirements listed in Chapter 1, some problems were handled during the developing processes. In order to make up the disadvantages of the current interface, more suitable downsample methods are used to avoid smoothing the peaks of datasets caused by data aggregation. The problem of slow plotting speed was solved by dynamically selected downsample rate. Besides, now timeline interaction makes it possible for different data types interacting with each other. The other expectation is to analyze the seismic stream, this goal was achieved by including the tools to visualize FFT and spectrogram in our interface. And with the use of a range slider, the user can shift among different time-lapse images flexibly.

This visualization tool can satisfy all requirements, and it will help the researchers get a better understanding of geographical phenomena from geophysical data.

But, there is still something can be completed to further improve this visualization tool:

- Implementing a mechanism to keep the preprocessed files up to date.
- Finding a suitable metric which can measure the downsampled quality from more objective aspects.
- Adopting an easier method to add the datasets from other sensors.
- Allowing the use of self-defined analysis methods.

Bibliography

[1] RL Hermanns, LH Blikra, M Naumann, B Nilsen, KK Panthi, D Stromeyer, and O Longva. Examples of multiple rock-slope collapses from köfels (ötz valley, austria) and western norway. *Engineering Geology*, 83(1-3):94–108, 2006.

[2] Antonio Contino, Patrizia Bova, Giuseppe Esposito, Ignazio Giuffré, and Salvatore Monteleone. Historical analysis of rainfall-triggered rockfalls: the case study of the disaster of the ancient hydrothermal sclafani spa (madonie mts, northern-central sicily, italy) in 1851. *Natural Hazards and Earth System Sciences*, 17:2229–2243, 2017.

[3] J Noetzli, M Hoelzle, and W Haeberli. Mountain permafrost and recent alpine rock-fall events: a gis-based approach to determine critical factors. In *Proceedings of the 8th International Conference on Permafrost*, volume 2, pages 827–832. Swets & Zeitlinger Lisse, Zürich, 2003.

[4] The PermaSense Project Group. PermaSense. `https://www.permasense.ch/en.html`. Accessed 2019-01-01.

[5] The PermaSense Project Group. PermaSense: GSN. `https://data.permasense.ch/data.html#data`. Accessed 2019-01-01.

[6] Meyer Matthias and Weber Samuel. PermaSense Python3 data toolbox(In prep.).

[7] Matthias Keller, Jan Beutel, Olga Saukh, and Lothar Thiele. Visualizing large sensor network data sets in space and time with vizzly. In *Local Computer Networks Workshops (LCN Workshops), 2012 IEEE 37th Conference on*, pages 925–933. IEEE, 2012.

[8] Sisense Contributors. Dash by plotly. `https://www.sisense.com`. Accessed 2019-01-01.

[9] Dash Contributors. Dash by plotly. `https://plot.ly/products/dash/`. Accessed 2019-01-01.

[10] Sveinn Steinarsson. Downsampling Time Series for Visual Representation. *MSc thesis, University of Iceland*, 2013.

[11] Jack Viljoen. LTTB algorithm for downsampling time series (Numpy implementation). `https://github.com/javiljoen/lttb.py`. Accessed 2019-01-01.

[12] obspy Contributors. WaveformPlotting. `https://github.com/obspy/obspy/blob/06906f389bdaeb5fa4c4bd0cab9f066082e7da42/obspy/imaging/waveform.py`. Accessed 2019-01-01.

[13] Peter Sammonds. Theoretical Seismology–Signal Processing. `http://www.ucl.ac.uk/EarthSci/people/sammonds/9%20Signal%20Processing.pdf`. Accessed 2019-01-01.

[14] The SciPy community. Discrete Fourier Transform (numpy.fft). `https://docs.scipy.org/doc/numpy/reference/routines.fft.html`. Accessed 2019-01-01.

[15] Carlos I Huerta-Lopez, Yongjune Shin, Edward J Powers, and Jose M Roesset. Time-frequency analysis of earthquake records. In *12th World Conference on Earthquake Engineering, Auckland*, 2000.

[16] The SciPy community. scipy.signal.spectrogram. `https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.spectrogram.html`. Accessed 2019-01-01.

[17] A.Gandhi Swati and kulkarnii C.V. Mse vs ssim. *International Journal of Scientific & Engineering Research*, 4:5, 2013.

[18] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.